Google™ **Web** Images Groups News Froogle **more »**

java 1.2 delegation-style custom class load Search Advanced Search Preferences

Web Results **1** - **3** of about **4** for java **1.2** delegation-style custom class loader. (**0.77** seconds)

Tip: Try Google Answers for help from expert researchers

Create a **custom Java 1.2**-style ClassLoader
... method to contain the **custom class**-loading logic **...** a no-op awaiting implementation by a **custom** ClassLoader **...** Create a **Java 1.2-delegation-style** ClassLoader Now I'll **...**
www.**java**world.com/**java**world/ jw-03-2000/jw-03-**class**load.html - Similar pages

[PDF] Pluggable Verification Modules: An Extensible Protection Mechanism ...
File Format: PDF/Adobe Acrobat - View as HTML
... PVMs to create a **custom**-made JVM plat **...** dynamic linking, access control, **delegation style** classloading, loading **...** to verification (eg, the **class java**.lang.Throwable **...**
www2.cs.uregina.ca/~pwlfong/Pub/oopsla2004.pdf - Similar pages

[PS] PROOF LINKING: A MODULAR VERIFICATION ARCHITECTURE FOR
File Format: Adobe PostScript - View as Text
... 9 **1.2** Thesis Statement . . **...** xiii. 5.1 **Java Delegation Style** Classloading . . **... Java** programs are composed of classes, each being loaded into the JVM separately. **...**
www2.cs.uregina.ca/~pwlfong/Pub/dissertation.ps - Similar pages

In order to show you the most relevant results, we have omitted some entries very similar to the 3 already displayed.
If you like, you can repeat the search with the omitted results included.

java 1.2 delegation-style custom Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

©2004 Google

9-13-2004

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.8

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review

Quick Links

» Se

Welcome to IEEE Xplore®

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

IEEE Enterprise

- Access the IEEE Enterprise File Cabinet

Your search matched **2** of **1071730** documents.
A maximum of **500** results are displayed, **50** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

dependency<and>class <and>load* Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **JMangler - a framework for load-time transformation of Java class f**
Kniesel, G.; Costanza, P.; Austermann, M.;
Source Code Analysis and Manipulation, 2001. Proceedings. First IEEE Interna Workshop on , 10 Nov. 2001
Pages:98 - 108

[Abstract] [PDF Full-Text (128 KB)] **IEEE CNF**

2 **Adaptive handoff channel management schemes for cellular mobile communication systems**
Kim, S.; Znati, T.F.;
Communications, 1999. ICC '99. 1999 IEEE International Conference on , Vol 2 , 6-10 June 1999
Pages:765 - 769 vol.2

[Abstract] [PDF Full-Text (472 KB)] **IEEE CNF**

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

9-13-2004

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.8

Welcome
United States Patent and Trademark Office

» Se

Help FAQ Terms IEEE Peer Review

Quick Links

Welcome to IEEE Xplore®

○ Home
○ What Can I Access?
○ Log-out

Tables of Contents

● Journals & Magazines
○ Conference Proceedings
○ Standards

Search

○ By Author
○ Basic
○ Advanced

Member Services

○ Join IEEE
○ Establish IEEE Web Account
○ Access the IEEE Member Digital Library

IEEE Enterprise

○ Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **4** of **1071730** documents.
A maximum of **500** results are displayed, **50** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

delegation<and>class<and>load* Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **Secure Java class loading**
Li Gong;
Internet Computing, IEEE , Volume: 2 , Issue: 6 , Nov.-Dec. 1998
Pages:56 - 61

[Abstract] [PDF Full-Text (64 KB)] **IEEE JNL**

2 **Implementing dynamic language features in Java using dynamic co generation**
Breuel, T.M.;
Technology of Object-Oriented Languages and Systems, 2001. TOOLS 39. 39 International Conference and Exhibition on , 29 July-3 Aug. 2001
Pages:143 - 152

[Abstract] [PDF Full-Text (480 KB)] **IEEE CNF**

3 **Information modeling of applications using mobile management ag with extensible behavior in run-time**
Randic, M.; Kunstic, M.; Blaskovic, B.;
Electrotechnical Conference, 2000. MELECON 2000. 10th Mediterranean , Vol 1 , 29-31 May 2000
Pages:15 - 18 vol.1

[Abstract] [PDF Full-Text (340 KB)] **IEEE CNF**

4 **Intelligent agent platform for procurement**
Subramanian, K.; Lee, S.; Tey Kar Shiang; Gan Beng Sue;
Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings IEEE International Conference on , Volume: 3 , 12-15 Oct. 1999
Pages:107 - 112 vol.3

7-13-04

[Abstract] [PDF Full-Text (596 KB)] IEEE CNF

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

A formal specification of Java class loading

Full text Pdf (241 KB)

Source **Conference on Object Oriented Programming Systems Languages and Applications** archive
Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications table of contents
Minneapolis, Minnesota, United States
Pages: 325 - 336
Year of Publication: 2000
ISBN:1-58113-200-X
Also published in ...

Authors
Zhenyu Qian Kestrel Institute, 3260 Hillview Avenue, Palo Alto, CA
Allen Goldberg Shoulders Corp., 800 West El Camino Real, Mountain View, CA and Kestrel Institute, 3260 Hillview Avenue, Palo Alto, CA
Alessandro Coglio Kestrel Institute, 3260 Hillview Avenue, Palo Alto, CA

Sponsor SIGPLAN: ACM Special Interest Group on Programming Languages

Publisher ACM Press New York, NY, USA

Additional Information: abstract references citings index terms collaborative colleagues peer to peer

Tools and Actions: Discussions Find similar Articles Review this Article
Save this Article to a Binder Display in BibTex Format

DOI Bookmark: Use this link to bookmark this Article: http://doi.acm.org/10.1145/353171.353193
What is a DOI?

↑ ABSTRACT

The Java Virtual Machine (JVM) has a novel and powerful mechanism to support lazy, dynamic class loading according to user-definable policies. Class loading directly impacts type safety, on which the security of Java applications is based. Conceptual bugs in the loading mechanism were found in earlier versions of the JVM that lead to type violations. A deeper understanding of the class loading mechanism, through such means as formal analysis, will improve our confidence that no additional bugs are present.The work presented in this paper provides a formal specification of (the relevant aspects of) class loading in the JVM and proves its type safety. Our approach to proving type safety is different from the usual ones since classes are dynamically loaded and full type information may not be statically available. In addition, we propose an improvement in the interaction between class loading and bytecode verification, which is cleaner and enables lazier loading.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 E. Borger and W. Schulte. Modular design for the Java virtual machine architecture.

ftp://ftp.di.unipi.it/pub/Papers/boerger/ jvmarch.ps, 1999.

2 G. Bracha. A critique of 'Security and dynamic loading in Java: A formalisation'. http://java.sun.com/people/gbracha /critique-jmt.html, 1999.

3 Luca Cardelli, Program fragments, linking, and modularization, Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.266-277, January 15-17, 1997, Paris, France

4 A. Coglio and A. Goldberg. Type safety in the JVM: Some problems in JDK 1.2.2 and proposed solutions. In Proc. ECOOP Workshop on Formal Techniques for Java Programs, 2000. Long version available at http://www.kestrel.edu/java.

5 A. Coglio, A. Goldberg, and Z. Qian. Towards a provably-correct implementation of the JVM bytecode verifier. In Proc. OOPSLA'98 Workshop Formal Underpinnings of Java, 1998.

6 Drew Dean, The security of static typing with dynamic linking, Proceedings of the 4th ACM conference on Computer and communications security, p.18-27, April 01-04, 1997, Zurich, Switzerland

7 S. Drossopoulou. Towards an abstract model of Java dynamic linking and verification. Department of Computing, Imperial College, London, UK.

8 Philip W. L. Fong , Robert D. Cameron, Proof linking: an architecture for modular verification of dynamically-linked mobile code, Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering, p.222-230, November 01-05, 1998, Lake Buena Vista, Florida, United States

9 Formal Methods Program - SRI Computer Science Laboratory. The PVS specification and verification system. http://pvs.csl.sri.com/, 1999.

10 Neal Glew , Greg Morrisett, Type-safe linking and modular assembly language, Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.250-261, January 20-22, 1999, San Antonio, Texas, United States

11 Allen Goldberg, A specification of Java loading and bytecode verification, Proceedings of the 5th ACM conference on Computer and communications security, p.49-58, November 02-05, 1998, San Francisco, California, United States

12 Li Gong, Inside Java 2 platform security architecture, API design, and implementation, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999

13 James Gosling , Bill Joy , Guy L. Steele, The Java Language Specification, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1996

14 T. Jensen , D. Le Métayer , T. Thorn, Security and Dynamic Class Loading in Java: A Formalization, Proceedings of the 1998 International Conference on Computer Languages, p.4, May 14-16, 1998

15 Sheng Liang , Gilad Bracha, Dynamic class loading in the Java virtual machine, Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, p.36-44, October 18-22, 1998, Vancouver, British Columbia, Canada

16 Tim Lindholm , Frank Yellin, Java Virtual Machine Specification, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999

17 Zhenyu Qian, Standard fixpoint iteration for Java bytecode verification, ACM Transactions on Programming Languages and Systems (TOPLAS), v.22 n.4, p.638-672, July 2000

18 Zhenyu Qian, A Formal Specification of Java Virtual Machine Instructions for Objects, Methods and Subrountines, Formal Syntax and Semantics of Java, p.271-312, January 1999

19 Z. Qian, A. Goldberg, and A. Coglio. A formal specification of JavaTM class loading. Long version. http://www.kestrel.edu/java, 2000.

20 V. Saraswat. Java is not type-safe. Technical report, AT&T Research, 1997. http://www.research.att.com/~vj/bug.html.

21 A. Tozawa and M. Hagiya. Careful analysis of type spoofing. In JIT'99 Java-Informations-Tage 1999, Clemens H. Cap, Hrsg., Informatik aktuell, pages 290-296. Springer Verlag, 1999.

22 A. Tozawa and M. Hagiya. New formalization of the JVM. http://nicosia.is.s.u-tokyo.ac.jp/members /miles/papers/cl-99.ps, 1999.

↑ CITINGS *4*

Sonia Fagorzi , Elena Zucca , Davide Ancona, Modeling multiple class loaders by a calculus for dynamic linking, Proceedings of the 2004 ACM symposium on Applied computing, March 14-17, 2004, Nicosia, Cyprus

Marcel Winandy , Armin B. Cremers , Hanno Langweg , Adrian Spalka, Protecting Java component integrity against Trojan Horse programs, Integrity and internal control in information systems V, Kluwer Academic Publishers, Norwell, MA, 2003

Gleb Naumovich, A conservative algorithm for computing the flow of permissions in Java programs, ACM SIGSOFT Software Engineering Notes, v.27 n.4, July 2002

Pieter H. Hartel , Luc Moreau, Formalizing the safety of Java, the Java virtual machine, and Java card, ACM Computing Surveys (CSUR), v.33 n.4, p.517-558, December 2001

↑ INDEX TERMS

Primary Classification:

D. Software
↪ **D.3** PROGRAMMING LANGUAGES
↪ **D.3.2** Language Classifications
↪ **Nouns:** Java

Additional Classification:

D. Software
↪ **D.3** PROGRAMMING LANGUAGES
↪ **D.3.4** Processors
↪ **Subjects:** Compilers

F. Theory of Computation

↪ **F.3** LOGICS AND MEANINGS OF PROGRAMS

General Terms:
Design, Languages, Performance, Theory

↑ **Collaborative Colleagues:**

Alessandro Coglio: Alessandro Armando
Antonio Camurri
Fausto Giunchiglia
Allen Goldberg
José Meseguer
Zhenyu Qian
Silvio Ranise
Carolyn L. Talcott

Allen Goldberg: Alessandro Coglio
Gordon Kotik
Peter Mills
Peter H. Mills
Lars S. Nyland
Robert Paige
Ira Pohl
Jan Prins
Jan F. Prins
Xiaolei Qian
Zhenyu Qian
John H. Reif
Robert A. Wagner
T. C. Wang
Tie-Cheng Wang
David Zimmerman

Zhenyu Qian: Alessandro Coglio
Régis Curien
Allen Goldberg
Hans-Jörg Kreowski
Bernd Krieg-Brückner
Tobias Nipkow
Hui Shi
Kang Wang

↑ **Peer to Peer - Readers of this Article have also read:**

- Data structures for quadtree approximation and compression
  Communications of the ACM 28, 9
  Hanan Samet

- A hierarchical single-key-lock access control using the Chinese remainder theorem
  Proceedings of the 1992 ACM/SIGAPP Symposium on Applied computing
  Kim S. Lee , Huizhu Lu , D. D. Fisher

- 3D representations for software visualization
  Proceedings of the 2003 ACM symposium on Software visualization
  Andrian Marcus , Louis Feng , Jonathan I. Maletic

- Probabilistic surfaces: point based primitives to show surface uncertainty
  Proceedings of the conference on Visualization '02
  Gevorg Grigoryan , Penny Rheingans

- Efficient simplification of point-sampled surfaces
  Proceedings of the conference on Visualization '02

Mark Pauly , Markus Gross , Leif P. Kobbelt

↑ **This Article has also been published in:**

- **ACM SIGPLAN Notices**
  Volume 35 , Issue 10 (October 2000)